

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA DEL SOFTWARE

CLASIFICACIÓN FACIAL EN GÉNERO Y EDAD
GENDER AND AGE CLASSIFICATION BASED ON FACIAL
FEATURES

Realizado por
Sergio Domínguez Fernández

Tutorizado por
Jesús Martínez Cruz
Rebeca Marfil Robles

Departamento
Lenguajes y CC. De la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, Julio de 2014

Fecha defensa:
El Secretario del Tribunal

Resumen:

El análisis facial es uno de los procesos claves para las interacciones humanas en la vida diaria. La gente es capaz de inferir el género, la edad, las emociones, etc. desde una única imagen de un rostro humano. También son capaces de reconocer con precisión a las personas que han conocido antes a pesar de los cambios en la iluminación, oclusiones parciales o cambios de perspectiva.

Un sistema capaz de detectar automáticamente los rostros y extraer datos de estas caras, sería muy útil para muchas aplicaciones diferentes: la seguridad, la vigilancia, la publicidad personalizada, las interfaces hombre-máquina, etc.

Este trabajo describe un sistema de clasificación de género y edad basado en el análisis facial. El sistema es capaz de detectar automáticamente y procesar rostros humanos para extraer estos datos, utilizando las siguientes etapas:

- La detección de rostros humanos en el entorno. Sólo los rostros que estén mirando a la cámara serán considerados para la clasificación, aunque se permite un cierto grado de cambio de perspectiva.
- Los rostros detectados son normalizados a un tamaño y perspectiva estándar.
- Los rostros normalizados son caracterizados utilizando sus vectores LBP.
- Seis SVM entrenados previamente serán utilizados para determinar el género y el rango de edad de la persona empleando el LBP.

El sistema ha sido probado usando un dispositivo Microsoft Kinect para capturar imágenes de entrada, y la Microsoft Kinect SDK para detectar y rastrear rostros. La biblioteca OpenCV también se ha empleado para normalizar las imágenes de la cara. Los seis clasificadores SVM han sido entrenados usando imágenes de la cara de la base de datos FERET. Por último, todo el sistema se ha integrado en RoboComp, un framework que utiliza Ice para comunicar los diferentes componentes del software. Los resultados muestran que el sistema es capaz de clasificar correctamente el 81% de las personas detectadas, a una velocidad de 12 imágenes por segundo en un PC estándar y usando secuencias de vídeo sin restricciones como entrada.

Palabras Claves: Visión por ordenador, análisis facial, clasificación de género, clasificación de edad, LBP, SVM.

Abstract:

Face analysis is one of the key processes for daily life human interactions. People are able to infer gender, age, emotions, etc. from a single image of a human face. They are also able to accurately recognize people they have met before despite changes in illumination, partial occlusions or perspective changes.

A system able to automatically detect faces and extract data from these faces would be very useful for many different applications: security, surveillance, personalized advertising, human-computer interfaces, etc.

This work describes a gender and age classification system based on facial analysis. The system is able to automatically detect and process human faces to extract these data, using the following stages:

- Detection of human faces in the environment. Only faces that are looking to the camera will be considered for classification, although a certain degree of perspective change is allowed.
- Detected faces are normalized to a standard size and perspective.
- Normalized faces are characterized using their LBP (Local Binary Patterns) vectors.
- Six trained SVM (Support Vector Machine) classifiers employ the LBP vector to set the gender and age range of the person.

The system has been tested using a Microsoft Kinect device to capture input images, and the Microsoft Kinect SDK to detect and track faces. The OpenCV library has also been employed to normalize face images. The six SVM classifiers have been trained using face images from the FERET database. Finally, the whole system has been integrated into RoboComp, a framework that uses Ice to communicate different software components. Results show that the system is able to correctly classify 81% of detected people, running at 12 frames per second in a standard PC and using unconstrained video sequences as input.

Keywords: Computer vision, Face analysis, Gender classification, Age classification, LBP, SVM.

ÍNDICE DE CONTENIDO

Capítulo 1.	1
1. Introducción	1
1.1.- Motivación.	1
1.2.- Objetivos.	3
1.3.- Estudio del arte.	5
1.3.1.- Un poco de historia.	6
1.3.2.- Algoritmos de Clasificación y Regresión.	10
1.4.- Estructura de la memoria.	12
Capítulo 2.	13
2. Esquema general del sistema de clasificación.	13
Capítulo 3	17
3. Extracción de características LBP.	17
3.1.- Normalización de las imágenes	21
Capítulo 4	23
4. Support Vector Machine.	23
4.1- Selección de parámetros y entrenamiento.	26
Capítulo 5	33
5. Resultados.	33
5.1.- Resultados obtenidos con imágenes de la base de datos.	33
5.2.- Resultados obtenidos con imágenes reales a través del sistema.	36
5.2.1.- Entorno de trabajo y arquitectura del sistema.	36
5.2.1.1. Kinect para Windows.	36
5.2.1.2. Arquitectura del sistema.	38
5.2.1.3. Ordenadores de sobremesa.	38
5.2.1.4. Robocomp.	39
5.2.2.- Resultados.	40
Capítulo 6	43

6. Conclusiones.....	43
6.1 Conclusiones.....	43
6.2 Trabajo futuro.....	44

Capítulo 1.

1. Introducción

1.1.- Motivación.

En nuestro día a día, normalmente no nos llama la atención de manera significativa nuestra capacidad para reconocer caras, incluso de individuos a los que no vemos desde hace bastante tiempo. Ya puedan tener distintas edades, o ser de distintas razas, o estar más lejos o más cerca, con mala o buena iluminación, siempre hemos tenido la capacidad de reconocer una cara atendiendo a nuestra experiencia previa o recuerdos. De hecho, el ser humano es capaz de reconocer una cara familiar desde prácticamente el primer mes de vida.

Sin embargo, esta particularidad no es tan simple cuando se intenta llevar a cabo con medios computacionales, utilizando herramientas hardware y software que presentan limitaciones y que, en buena parte, dependen de la calidad que disponga la información que se haya recogido respecto al rostro que se quiera identificar y de la metodología utilizada.

Desde los inicios de la visión por ordenador, el reconocimiento de rostros es considerado como uno de los problemas que son dignos de estudiar y profundizar. De hecho, éste es un tema que ha alcanzado mucho auge en los últimos años, propiciado por el avance de los sistemas informáticos particularmente en el cálculo computacional, que ha permitido desarrollar e implementar algoritmos cada vez más complejos utilizando diversas técnicas.

La identificación de características o rasgos faciales ha adquirido un fuerte impulso gracias al avance en la tecnología de video multimedia, consiguiéndose así un aumento de cámaras en lugares como el trabajo y el hogar a un precio reducido.

El reconocimiento facial se puede aplicar en diversas situaciones tales como el control de acceso a edificios (ya sean públicos o privados), a cajeros automáticos, a laboratorios de investigación y últimamente también como medio de acceso para el uso de ordenadores personales o terminales móviles de última generación. Utilizando estos sistemas, se podrían crear bases de datos con las personas que tienen acceso a lugares concretos como los mencionados anteriormente.

Básicamente, el proceso de identificación facial se divide en dos tareas: una **detección** previa y un **reconocimiento** posterior. La primera de ellas, la detección, comprende la localización de una cara dentro de una imagen, mientras que la segunda tarea, la del reconocimiento, consiste en la comparación de la cara detectada en la primera fase con otras ya almacenadas en una base de datos.

Las dos tareas antes mencionadas, la de detección y la de reconocimiento, deberían estar relacionadas entre sí de alguna manera, ya que según la forma en la que se detecte una cara en la primera etapa, puede resultar imposible su reconocimiento en la segunda etapa con caras de una base de datos que hayan sido detectadas de forma distinta, de ahí que los sistemas de reconocimiento facial estén muy condicionados por la posición y orientación de la cara del individuo con respecto a la cámara y también por las condiciones de iluminación existentes en el momento de realizar la captura de las imágenes.

1.2.- Objetivos.

El objetivo de este trabajo fin de grado no es otro que el de crear un software de reconocimiento facial capaz de clasificar las personas por género y edad.

Frente a otras posibilidades, el sistema se basa en la captura de una imagen facial frontal del individuo a clasificar, que mediante un proceso de normalización es preprocesada para conseguir que en todas las imágenes se capture el mismo porcentaje de área facial y con el mismo tamaño y resolución. Ésta imagen preprocesada es caracterizada usando métodos para imagen 2D (en concreto se usará el LBP). La imagen facial es capturada utilizando la SDK de Microsoft Kinect que permite el seguimiento en tiempo real de la cara del sujeto y que proporciona los ángulos de giro de la misma, permitiendo estimar si la cara del sujeto está situada frontal a la cámara. La SDK de Kinect proporciona además la posición de los ojos del sujeto que serán utilizados en el proceso de normalización.

Por lo tanto, el sistema propuesto no es invariante a la pose de la cara del sujeto a clasificar. Pese a todo esto, el ciclo de trabajo seguirá un proceso iterativo de diseño-evaluación. Si la evaluación no ofrece los resultados deseados, este diseño podría ser modificado para incluir características invariantes.

El sistema que se describe en este Trabajo Fin de Grado trabajará, específicamente, con los siguientes grupos de clasificación:

- ❖ Niños (<15)
- ❖ Mujeres jóvenes (15-40)
- ❖ Hombres jóvenes (15-40)
- ❖ Mujeres adultas (40-60)
- ❖ Hombres adultos (40-60)
- ❖ Mujeres mayores (> 60)
- ❖ Hombres mayores (>60)

Para este problema de clasificación, se ha optado por el empleo de los patrones locales binarios (Local Binary Patterns, LBP) como técnica de caracterización facial, que serán clasificados posteriormente por un algoritmo de vectores de soporte (Support Vector

Machine, SVM) previamente entrenado para discriminar entre los grupos anteriormente presentados.

Los LBP son un descriptor de textura. La idea es representar la estructura local de la imagen comparando cada píxel con sus vecinos. Se toma cada píxel como centro de una ventana y se compara con sus vecinos en esta ventana: si el vecino tiene una intensidad mayor que la suya se codifica con un 1, si es menor con un 0. De esta forma, por cada píxel de la imagen se tiene un número binario denominado LBP. Los LBPs de toda la imagen se agrupan obteniendo un histograma.

El Support Vector Machine (SVM) es un clasificador discriminativo definido formalmente por un hiperplano de separación. En otras palabras, dado un conjunto de características de entrenamiento debidamente etiquetadas (aprendizaje supervisado), el algoritmo es capaz de calcular un hiperplano óptimo que separa las diferentes categorías o grupos de clasificación, haciendo posible clasificar nuevos datos de entrada.

Este algoritmo permite la clasificación tanto de datos separables de forma lineal como de datos no separables linealmente; para este último caso, se ha de definir un kernel que permita mapear los datos no separables linealmente en un conjunto de datos separables linealmente, aumentando su dimensión.

El método de clasificación utilizado consta, por tanto, de dos etapas claramente diferenciadas:

- ❖ Una primera etapa de entrenamiento o aprendizaje supervisado, que sólo se realiza una vez, offline, con un conjunto de imágenes pregrabadas de entrenamiento.
- ❖ Una etapa de funcionamiento normal, en la que nuevas imágenes de entrada son clasificadas online en una de las categorías anteriormente presentadas.

1.3.- Estudio del arte.

La Real Academia Española define la **biometría** como el “estudio mensurativo o estadístico de los fenómenos o procesos biológicos”. Sin embargo, podemos definirla de forma más precisa como el conjunto de métodos automatizados que analizan determinadas características humanas para identificar o autenticar personas en función de determinados rasgos físicos (cara, voz, retina, iris, huellas dactilares, etc.) o conductuales (forma de escribir o de andar entre otras).

Dentro de la biometría, podemos hablar de dos finalidades u objetivos concretos [1] como son:

- La verificación biométrica, consistente en usar una determinada tecnología para asegurar que el individuo que debemos reconocer es quien dice ser.
- La identificación biométrica, en la que no queremos confirmar que un usuario es quien asegura ser, sino que tratamos de aplicar la tecnología para conocer la identidad de un individuo.

En los últimos años se ha avanzado mucho en los sistemas de reconocimiento, tanto en las técnicas clásicas como las huellas dactilares o el reconocimiento por voz, como en las más novedosas en las que se utilizan rasgos de nuestra cara para el reconocimiento del individuo; estos sistemas reciben el nombre de **biometría facial**.

Este avance se debe principalmente a las novedades presentadas en este sector del reconocimiento y a la necesidad de las propias administraciones por adaptar estas técnicas en sus controles policiales y de seguridad.

A diferencia de otras biometrías tipo iris o huella dactilar esta tecnología no es intrusiva y no necesita de colaboración por parte del usuario. Sólo es necesario que su rostro sea

adquirido por una cámara. Los puntos que se recogen sirven para elaborar un mapa de puntos que conforman, a través de unos algoritmos matemáticos, el “patrón”.

El sistema de reconocimiento facial ha realizado grandes avances y ya es posible reconocer a una persona aunque ésta lleve gafas, bigote, peluca, etc. La tecnología convencional de reconocimiento facial contiene puntos inherentes débiles como son la intensidad de la iluminación y la variación de la pose: dificultades si la imagen del rostro no es frontal, o si las condiciones de iluminación son inadecuadas [2].

Últimamente se está investigando en el reconocimiento tridimensional [3][4], con él se pretende tomar una imagen tridimensional de la cara de una persona y elaborar un patrón en base a esta toma, con lo cual, se busca que el reconocimiento se pueda realizar independientemente del ángulo de la toma.

1.3.1.- Un poco de historia.

Para localizar el origen de lo que actualmente llamamos reconocimiento facial hay que remontarse a los años sesenta, donde los primeros sistemas que aparecieron eran semiautomáticos, y eran capaz de reconocer, gracias a un administrador externo, rasgos como ojos, nariz, orejas o boca, para así, tomar medidas de distancias de referencias y compararlas con un patrón dado.

En los años 70 Goldstein, Harmon y Lesk [5], usaron 21 marcadores subjetivos específicos tales como el color del cabello y grosor de labios para automatizar el reconocimiento facial. El problema con estas soluciones previas era que se computaban manualmente.

En 1988 Kirby y Sirovich [6] aplicaron análisis de componentes principales, una técnica estándar del álgebra lineal, al problema del reconocimiento facial. Esto fue considerado algo así como un hito al mostrar que eran requeridos menos de 100 valores para cifrar acertadamente la imagen de una cara convenientemente alineada y normalizada.

En 1991, Turk y Pentland [7], utilizando las técnicas Eigenfaces, determinaron que el error residual podía ser utilizado para detectar caras en las imágenes, un descubrimiento que permitió sistemas automatizados de reconocimiento facial en tiempo real fidedignos. Si bien la aproximación era un tanto forzada por factores ambientales, creó sin embargo un interés significativo en posteriores desarrollos de éstos sistemas.

A partir de los noventa surge la biometría facial tal y como la entendemos hoy en día, aunque su implementación práctica no llegaría hasta enero de 2001, con la celebración de la Super Bowl de la NFL, donde se archivaron fotografías de los sistemas de vigilancia y se compararon con bases de datos digitales [8].

Esta demostración inició un muy requerido análisis sobre cómo usar la tecnología para satisfacer necesidades nacionales, mientras se tomaban en consideración las preocupaciones sociales y de privacidad del público.

Hoy, la tecnología de reconocimiento facial está siendo utilizada para combatir la falsificación de pasaportes, soporte al orden público, identificación de niños extraviados, minimizar el fraude en las identificaciones e incluso para el marketing.

Por último, la biometría facial también ha integrado en los últimos años sistemas de reconocimiento mediante vídeo [9]. El problema de utilizar estos sistemas de videovigilancia (habituales en controles de seguridad), es la baja calidad de las imágenes grabadas, así como el pequeño tamaño con el que se observan las caras en estos estudios.

La elección del vídeo frente a las imágenes, por otro lado, también presenta algunas ventajas, como el de realizar un seguimiento de las caras (ya que da cierta “continuidad temporal”), y por supuesto, el dar más información sobre las personas que estamos tratando de identificar o verificar (gestos, forma de andar, etc.).

En la actualidad se está trabajando por mejorar los análisis biométricos mediante innovaciones en el software utilizado [10]. Por ejemplo, hoy en día los algoritmos existentes tienen grandes problemas para trabajar con imágenes donde las caras estén giradas más de veinte grados. El reto de algunos de estos estudios es ser capaz de analizar

fotografías donde el rostro esté a casi cuarenta y cinco grados, lo que permitiría dar un salto importante en las investigaciones criminales por ejemplo.

Mejorar las resoluciones de las imágenes, además de otras características y desafíos, como por ejemplo que trabajemos con imágenes parciales o mal iluminadas, son varios de los problemas que ha de superar la industria de la tecnología del reconocimiento facial. Solo así, en casos como lo ocurrido en los atentados de Boston de 2013, la biometría facial dejará de ser una esperanza, para convertirse en una **realidad tecnológica** de gran impacto en la sociedad.

Si queremos realizar una clasificación según la edad y el género, además de los problemas ya mencionados en el proceso de detección de caras, nos vamos a encontrar con otros escollos a los que habrá que hacer frente. Uno de ellos es, por ejemplo, el distinto grado de envejecimiento facial que presenta el ser humano (figura 1), acuciado por factores tan diversos como los genéticos o los relacionados con la salud, estilo de vida o el propio clima.

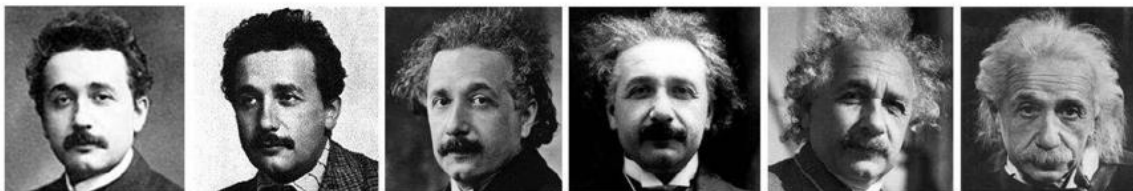


Figura1: Etapas del envejecimiento en una misma persona

Actualmente para estimar la edad de una cara de una imagen se realiza una primera etapa de representación de la edad por medio de técnicas como:

- *Modelos antropométricos.* Estos modelos fueron los primeros utilizados para estimar la edad a partir de imágenes faciales [11] [12], y se centran en los cambios de tamaño del cráneo a lo largo de los años [13] pero, aunque se suele obtener un 100 % de acierto, sólo pueden tratar con niños o jóvenes por ser en estas etapas de la vida en las que la forma de la cabeza presenta cambios significativos. Además, esta técnica sólo trabaja con imágenes faciales frontales y de alta resolución.

- *Modelos activos de apariencia (Active Appearance Models AAM)*. Esta técnica construye un modelo general a partir de un conjunto de imágenes [14]. Dicho modelo intenta explicar la variabilidad de la forma y textura en dicho conjunto. De esta forma, cada imagen se puede interpretar como una variación del modelo general previamente construido. En AAM cada imagen es representada por un vector de datos en el que se incluyen las coordenadas de puntos de referencia como los ojos, la nariz, los labios, etc., y/o información local en esos puntos de referencia, como puede ser la textura (Figura 2). Dado un conjunto de imágenes faciales y los correspondientes puntos de referencia, se genera un modelo estático de la variación de forma utilizando análisis de componentes principales (Principal Component Analysis, PCA). De esta manera, se construye un modelo estático de variación de textura, siendo este modelo lo único necesario para extraer el vector de características. Los sistemas que utilizan estos modelos tienen que disponer de un etiquetamiento de los puntos de la cara, y por lo tanto, tienen que disponer de un sistema capaz de localizar esos puntos de referencia. Por lo tanto, el correcto funcionamiento del sistema, está estrechamente ligado al correcto localizador de los puntos de referencias.

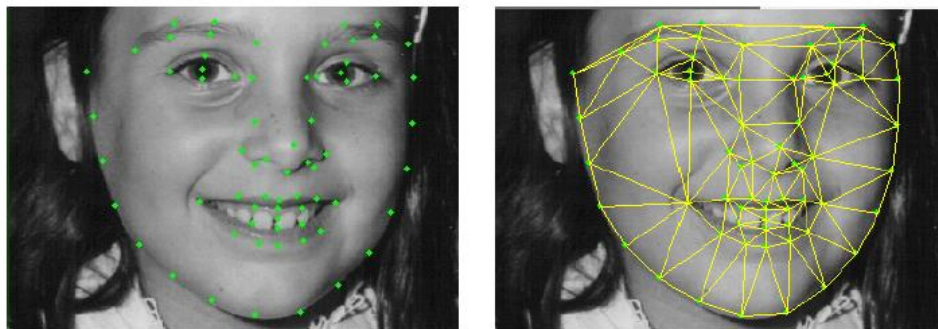


Figura 2: Puntos de referencia y vector que forma la unión de dichos puntos mediante triangulación de Delaunay (la circunferencia circunscrita de cada triángulo de la red no debe contener ningún vértice de otro triángulo) para AAM.

- *Subespacio de patrones de envejecimiento (Aging Pattern Subspace AGES)*, es una técnica que para modelar el patrón de envejecimiento considera de forma conjunta todas las imágenes de un individuo ordenadas en el tiempo en vez de forma separada para generar un subespacio representativo para cada edad [15]. La información que maneja este método es la proporcionada por AAM, pero con la

diferencia de que éste trabaja con secuencias de imágenes de la misma persona. El problema que presenta es que no siempre se tienen imágenes de todas las edades, y el patrón de envejecimiento obtenido no siempre es completo, aunque existen algoritmos para paliar esas carencias; además presenta el problema del localizador de puntos propios de AAM y también, que supone que todas las personas tienen las mismas características faciales a una determinada edad, lo cual es incierto en la vida real, ya que el proceso de envejecimiento es individual y está afectado por factores externos.

- *Age Manifold*. En lugar de calcular un patrón de envejecimiento por cada individuo de entrenamiento, se puede calcular uno común a muchos individuos y edades. Así, no es necesario que cada individuo de entrenamiento tenga imágenes a muchas edades para formar la base de datos. El principal requisito de estos sistemas es que se necesita una base de datos de entrenamiento bastante grande para que el patrón de envejecimiento se calcule correctamente.
- *Modelos de apariencia*. Se extraen características faciales teniendo en cuenta la textura y la forma de cada imagen facial. En [16] se usan Patrones Binarios Locales (LBP) para extraer características faciales, presentando unos resultados prometedores.

1.3.2.- Algoritmos de Clasificación y Regresión.

Si consideramos cada edad o rango de edades como una clase, el problema de estimación de las edades se puede ver como un problema de clasificación. Por otro lado, al ser las edades un conjunto de valores secuenciales, también podemos ver la estimación de la edad como un problema de regresión al querer predecir la edad concreta de un individuo.

Estimación de edad como problema de Clasificación

Se han evaluado el funcionamiento de diferentes clasificadores [17], incluyendo Nearest Neighbor (NN), Redes Neuronales Artificiales (RNA) y un clasificador de función

cuadrática. Aunque la función cuadrática es una función de regresión, que relaciona las representaciones de cara con etiquetas de edad, los autores la han denominado clasificador de función cuadrática. Las pruebas se han realizado en una base de datos pequeña (400 imágenes) con un rango de edades de 0 a 35 años. Los resultados obtenidos muestran que el clasificador de función cuadrática tiene un error absoluto medio de 5.04 años, este error es ligeramente menor del obtenido por el NN, pero más alto que el obtenido con RNAs.

Otros autores [18][19] utilizan *Support Vector Machine (SVM)* en una base de datos con 8000 imágenes. Los errores absolutos medios son de 5.55 y 5.52 años para mujeres y hombres, respectivamente.

Estimación de edad como problema de Regresión

También se han estudiado [20] tres funciones de regresión distintas: lineal, cuadrática y cúbica. Los experimentos realizados por estos autores muestran que utilizando las funciones cuadrática y cúbica los errores se reducen significativamente, comparados con los errores al utilizar la función lineal. Entre utilizar la función cuadrática y la cúbica no hay diferencias significativas, por lo que los autores recomiendan utilizar la función cuadrática por ser la más sencilla. Ciertos autores [21][22] usan una regresión lineal múltiple. Otros [23][24] utilizan el método de Support Vector Regression (SVR), con un error absoluto medio de 5.16.

En los párrafos anteriores se han comentado alguno de los métodos existentes de clasificación y regresión en problemas de estimación de edad, pero el enfoque que parece más acertado, como se muestra en [25][26][27], es combinar ambos enfoques: clasificación y regresión.

Para estimar la edad se utilizan métodos basados en características o métodos basados en apariencia. Los métodos más antiguos son los basados en apariencia, pero los utilizados en la actualidad son aquellos en los que predomina la extracción de características, llegando a alcanzar un porcentaje de clasificación del 96'75% usando SVM como clasificador y LBP para la caracterización previa de la cara [28]

1.4.- Estructura de la memoria.

La memoria está estructurada de la siguiente forma:

En el Capítulo 2 se presenta el *esquema general del sistema de clasificación* haciendo especial diferenciación en cada una de las dos etapas que lo forman; por un lado la detección de la cara produciendo así los LBP, y por otro lado, la clasificación de la cara detectada en un rango dado.

En el Capítulo 3 de *Extracción de características LBP*, se continúa con un estudio más detallado del descriptor de la cara finalmente seleccionado.

En el Capítulo 4 de *Support Vector Machine*, se describe el algoritmo de clasificación SVM.

En el Capítulo 5 de *Resultados*, se detallan los resultados obtenidos con las imágenes de la base de datos utilizada para entrenar el clasificador y con imágenes reales.

En el Capítulo 6 de *Conclusiones*, se revisan los objetivos propuestos y las conclusiones derivadas de los resultados. Además se plantean posibles trabajos futuros como continuación a este Trabajo Fin de Grado.

Capítulo 2.

2. Esquema general del sistema de clasificación.

El algoritmo de clasificación que se va a utilizar en este trabajo fin de grado tiene, desde el punto de vista de diseño, dos etapas completamente diferenciadas (Fig. 3):

- Extracción de características LBP de la imagen facial de entrada.
- Clasificación de las características extraídas en uno de los rangos comentados anteriormente utilizando SVM.

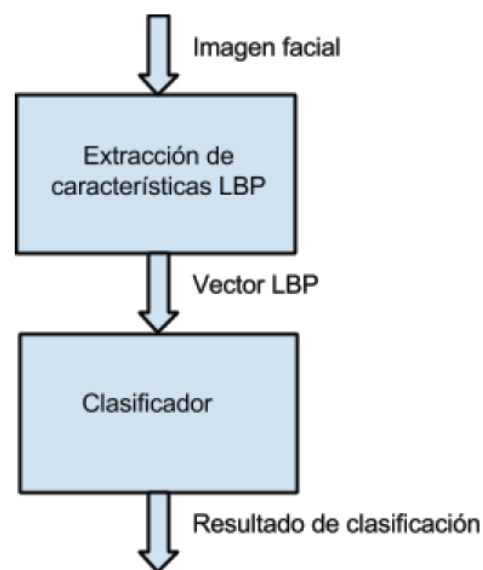


Figura 3: Etapas del algoritmo de clasificación

El algoritmo SVM ha demostrado funcionar mejor cuando los grupos de clasificación son sólo dos (clasificación binaria). Debido a esto, se ha optado por utilizar varios clasificadores en cascada dentro del bloque de clasificación, obteniendo el esquema de la Figura 4.

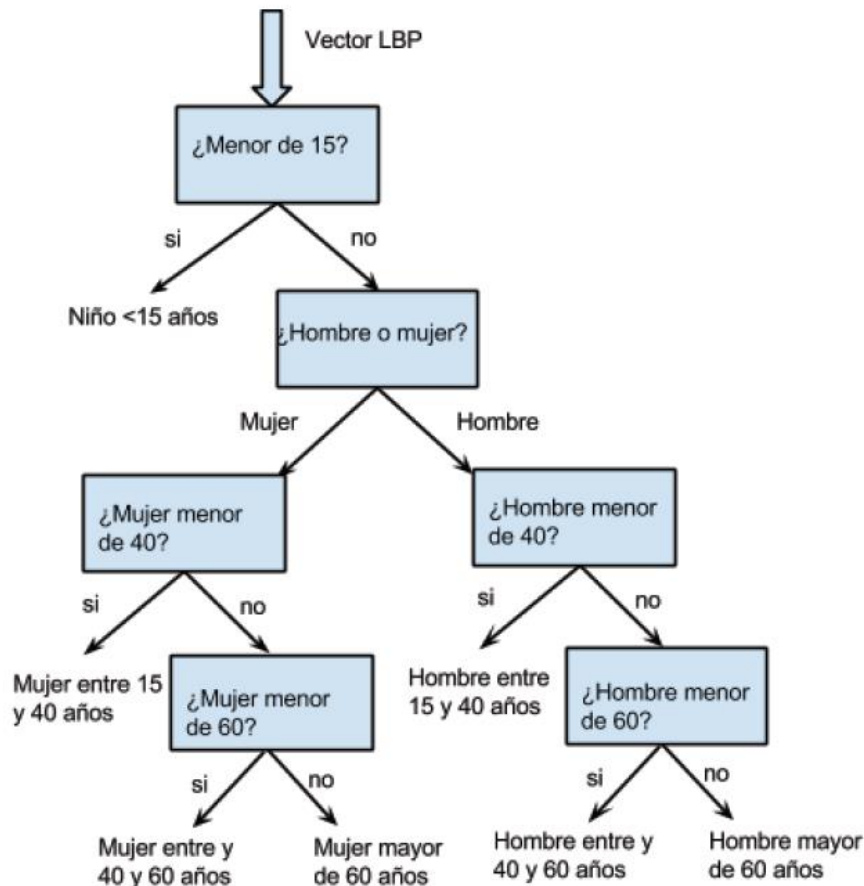


Figura 4: Esquema general de la etapa de clasificación

En este esquema, cada una de las preguntas que aparecen en las cajas se corresponde con un clasificador SVM entrenado de forma diferente. Así tendríamos seis clasificadores:

- **Clasificador 1:** Distingue entre menores de 15 años y mayores de 15 años.
- **Clasificador 2:** En el caso de ser clasificado como mayor de 15 años aparece un segundo clasificador que distingue entre hombres y mujeres mayores de 15 años.

- **Clasificador 3:** En el caso de ser clasificado como hombre el siguiente clasificador discrimina entre hombres mayores de 15 años pero menores de 40 y hombres mayores de 40 años.
- **Clasificador 4:** Finalmente los hombres mayores de 40 años pasan por un cuarto clasificador que distingue entre hombres mayores de 40 años pero menores de 60 y hombres mayores de 60 años.
- **Clasificador 5:** En el caso de que el clasificador 2 clasifique la entrada como mujer, el siguiente clasificador discrimina entre mujeres mayores de 15 años pero menores de 40 y mujeres mayores de 40 años.
- **Clasificador 6:** Finalmente las mujeres mayores de 40 años son clasificadas entre mujeres mayores de 40 años pero menores de 60 años y mujeres mayores de 60 años.

Desde el punto de vista del funcionamiento, este sistema de clasificación tiene dos etapas también: una etapa de entrenamiento o aprendizaje supervisado y una etapa de funcionamiento normal del algoritmo.

Capítulo 3

3. Extracción de características LBP.

Como ya se ha comentado anteriormente los *Local Binary Patterns* o LBP son unos descriptores de textura muy utilizados para reconocimiento facial.

En lugar de representar la imagen de entrada, en este caso una imagen facial, como un único vector de dimensión elevada, se representan características locales de textura con vectores de dimensión más reducida.

La idea básica es representar la estructura local de una imagen comparando cada píxel con sus vecinos. Se toma un píxel como centro y se compara con cada uno de sus vecinos. Si la intensidad de este píxel central es mayor o igual que la del vecino con el que se compara entonces se marca con un 1 y si no con un 0. Para el caso de vecindad 8 se obtendría, por ejemplo, para un píxel central un número binario del tipo 11001110. Se tendrían entonces 256 posibles combinaciones (2^8), denominadas códigos LBP. El más sencillo de obtener es aquel en el cual los vecinos están a una distancia 1 (radio 1) del píxel central, como en este caso, aunque se pueden coger radios mayores:

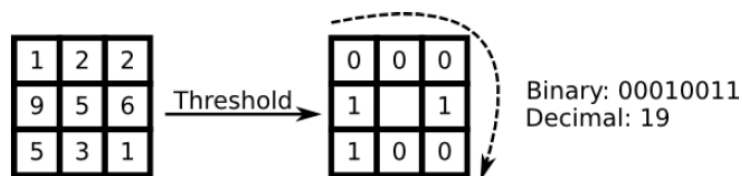


Figura 5: obtención del código LBP

Una descripción formal del operador LBP es la ofrecida por la ecuación siguiente:

$$\text{LBP}(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

donde (x_c, y_c) es el píxel central con intensidad i_c , y i_p es la intensidad del píxel vecino, P es el número de vecinos y $s(x)$ una función del tipo:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

La idea es poder usar radios distintos de uno, con lo cual tendríamos los vecinos situados en un círculo con el centro en el píxel de estudio. Así, dado un punto (x_c, y_c) , la posición del vecino (x_p, y_p) con $p \in P$ se puede calcular de la siguiente forma:

$$\begin{aligned} x_p &= x_c + R \cos\left(\frac{2\pi p}{P}\right) \\ y_p &= y_c - R \sin\left(\frac{2\pi p}{P}\right) \end{aligned}$$

siendo R el radio utilizado y P el número de vecinos. El operador es una extensión de los códigos LBP y a veces se llama LBP extendido o circular. Si un punto del círculo calculado de esta forma no se corresponde con un píxel de la imagen entonces se interpola. La implementación OpenCV utiliza como esquema de interpolación una interpolación bilineal:

$$f(x, y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}.$$

Por definición el operador LBP es robusto frente a cambios en escala de grises. En la siguiente imagen podemos apreciar claramente esta capacidad donde en la segunda fila vemos la imagen LBP de una imagen modificada artificialmente.

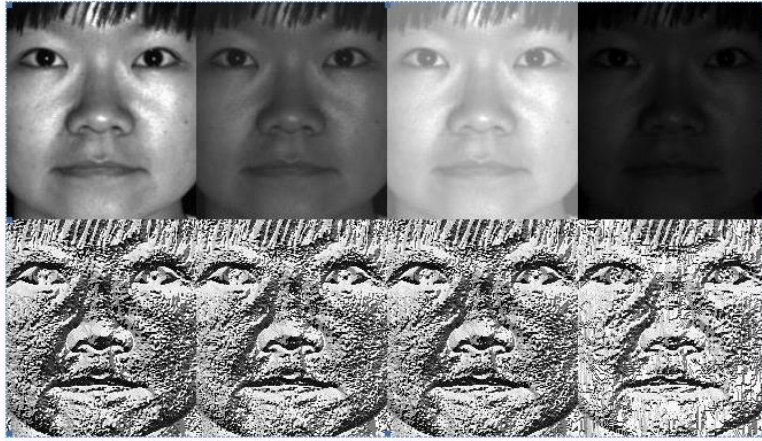


Figura 6: Robustez frente a cambios de escala de grises

Lo que hace el algoritmo de extracción de características LBP es dividir la imagen en m regiones; dentro de cada región se calculan los vectores de características correspondientes tomando cada uno de los píxeles como píxel central, y se calcula un histograma con todos ellos. De esta forma se tienen m histogramas, los cuales se concatenan, siendo esta concatenación el vector de características final de la imagen.

Para el cálculo de este vector de características LBP se ha utilizado la librería de visión por ordenador “OpenCV”. En concreto, en las pruebas de verificación del descriptor se han elegido los siguientes valores para los parámetros:

- $R=1$. (Radio 1).
- $m=64$ (la imagen de entrada se divide utilizando un grid de 8×8).
- $P=8$. (vecindad 8).

Con estos parámetros se obtiene un vector final de características de $2^8 \times 64 = 16384$ elementos o componentes por cada imagen.

A nivel global, el código es el siguiente:

```
Mat image;  
Ptr<FaceRecognizer> model; //Modelo LBP  
  
model = createLBPHFaceRecognizer(1,8,8,8);  
image=(imread("imagen_entrada.jpg",0));  
  
Mat hist=model->crear_histograma(image); //Calcula el vector de características LBP
```

Figura 7: Código para el cálculo del vector LBP

Para la obtención de buenos resultados utilizando LBP, es necesario realizar una normalización previa de las imágenes de entrada, que se explica a continuación.

3.1.- Normalización de las imágenes

Para un correcto funcionamiento del sistema de clasificación basado en LBP, es necesario que las imágenes faciales tengan todas el mismo tamaño, resolución y que en todas se abarque el mismo porcentaje de área facial. Para ello y haciendo uso de las posiciones de los ojos, OpenCV permite realizar el recorte y reajuste del tamaño de las imágenes mediante un script, con el que se obtiene el resultado que aparece en la figura 8.

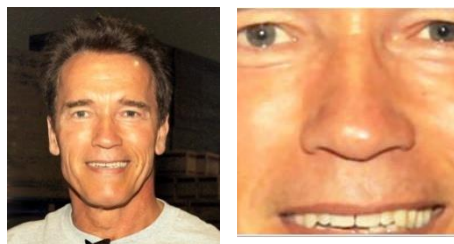


Figura 8: Imagen original e imagen normalizada.

Las posiciones de los ojos son calculadas utilizando la librería OpenCV, en el caso de trabajar con imágenes faciales almacenadas en ficheros, o bien las proporciona la kinect en el caso de trabajar con imágenes capturadas on-line.

Capítulo 4

4. Support Vector Machine.

La idea básica es que, dado un conjunto de puntos, subconjunto de un conjunto mayor (espacio), en el que cada uno de los puntos pertenece a uno de dos posibles grupos de clasificación o categorías, un algoritmo basado en SVM construye un modelo capaz de predecir si un punto nuevo (cuyo grupo de clasificación se desconoce) pertenece a una categoría u otra. Como en la mayoría de los métodos de clasificación supervisada, los datos de entrada (los puntos) son vistos como un vector p-dimensional.

SVM busca un hiperplano que separe (figura 9) de forma óptima a los puntos de una clase de la de otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior (si no son separables linealmente). En ese concepto de "separación óptima" es donde reside la característica fundamental del SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. Por eso también a veces se conoce a los SVM como *clasificadores de margen máximo*. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado.

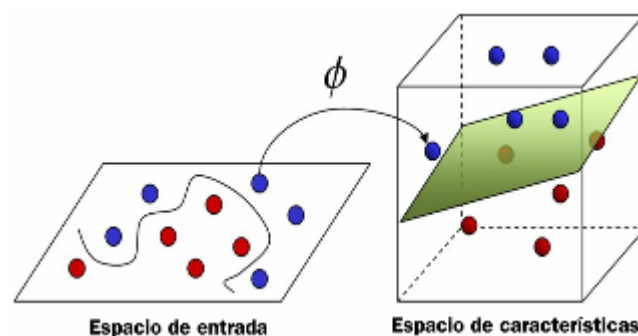


Figura 9: Hiperplano de separación.

En la literatura de los SVMs, se llama *atributo* a la variable predictora y *característica* a un atributo transformado que es usado para definir el hiperplano. La elección de la representación más adecuada del universo estudiado se realiza mediante un proceso denominado selección de características. Al vector formado por los puntos más cercanos al hiperplano se le llama vector de soporte, lo que da nombre al método (figura 10).

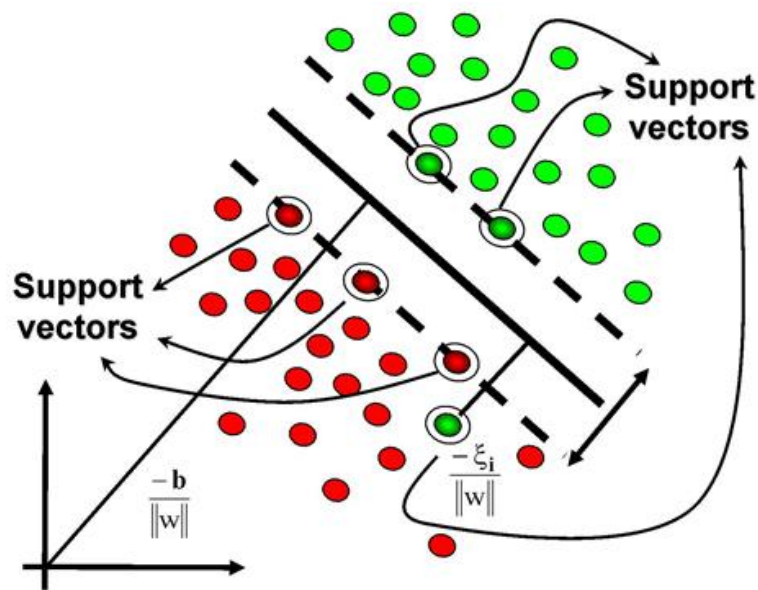


Figura 10: Obtención de los vectores de soporte.

Idealmente, el modelo basado en SVM debería producir un hiperplano que separe completamente los datos del universo estudiado en dos categorías. Sin embargo, no es siempre posible encontrar una separación perfecta y, si lo es, el resultado del modelo no puede ser generalizado para otros datos. Esto se conoce como sobreajuste (*overfitting*). Con el fin de permitir cierta flexibilidad, los SVM manejan un parámetro interno, C , que controla la compensación entre errores de entrenamiento y los márgenes rígidos, creando así un margen blando (*soft margin*) que permita algunos errores en la clasificación a la vez que los penaliza.

La manera más simple de realizar la separación entre categorías es mediante una línea recta, un plano recto o un hiperplano N dimensional.

Desafortunadamente los universos a estudiar (como es el caso de la clasificación en edad y sexo) no suelen ser separables linealmente. La representación por medio de funciones kernel ofrece una solución a este problema (figura 11), proyectando la información a un espacio de características de mayor dimensión.

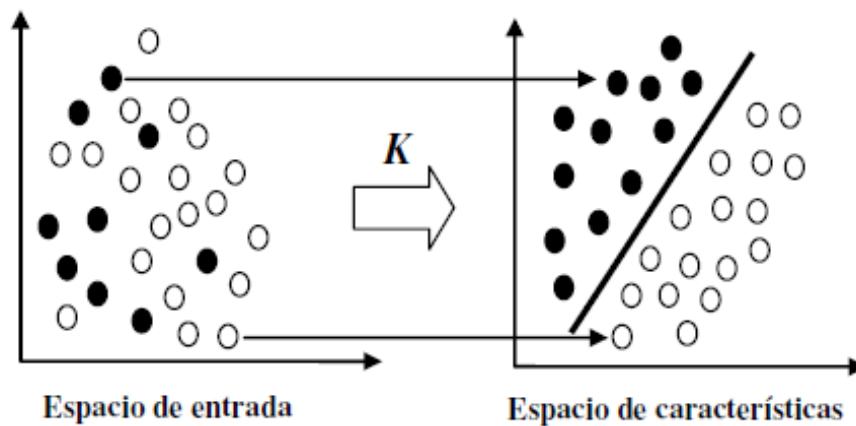


Figura 11: SVM no linealmente separable inducida por una función kernel.

Los tipos de kernel más utilizados son:

- Polinomial-homogénea

$$K(x_i, x_j) = (x_i \cdot x_j)_n$$

- Perceptron

$$K(x_i, x_j) = \|x_i - x_j\|$$

- Función de base radial Gaussiana

$$K(x_i, x_j) = \exp(-\gamma(x_i - x_j)^2)$$

- Tangente hiperbólica

$$K(x_i, x_j) = \tanh(Kx_i \cdot x_j + c)$$

Aunque los SVM permiten realizar una clasificación en N categorías, su funcionamiento es óptimo en el caso de tener que discriminar únicamente entre dos categorías. Debido a esto se ha optado por el esquema de la Figura 4 y no se ha utilizado un único clasificador.

4.1- Selección de parámetros y entrenamiento.

Teniendo en cuenta lo explicado anteriormente, la efectividad del SVM depende del kernel que se use, de los parámetros de dicho kernel y del parámetro C. Para el caso de la clasificación por edad y sexo se ha optado por la utilización del kernel de base radial gaussiana (RBF), que es el más utilizado en aplicaciones de procesamiento de imagen. En este caso, la mejor combinación de C y γ se ha obtenido realizando una búsqueda en grid y un proceso de validación cruzada.

La búsqueda en grid es una forma muy usada de realizar optimización de parámetros en algoritmos de aprendizaje supervisado y que consiste en realizar una búsqueda exhaustiva entre un subconjunto de parámetros especificados a mano. Para el caso del SVM con RBF se elige un conjunto de valores de γ y C, por ejemplo:

$$C = \{10, 100, 1000\}$$

$$\gamma = \{0.1, 0.2, 0.5, 1.0\}$$

La búsqueda en grid lo que hace es entrenar el SVM para cada pareja (C, γ) utilizando un conjunto de imágenes de entrenamiento, y evalúa su funcionamiento mediante un proceso de validación cruzada. Este proceso consiste en dividir las imágenes de entrenamiento en k subconjuntos, de forma que k-1 se usan para entrenar el sistema y el restante para validarlo. Se realizan k entrenamientos y validaciones, variando los subconjuntos utilizados en cada caso. Por ejemplo, supongamos que tenemos 15 imágenes de entrenamiento que dividimos en 5 subconjuntos con 3 imágenes cada uno, se realizarían cinco procesos de entrenamiento-validación variando los subconjuntos como aparece en la figura 12. En cada proceso de entrenamiento-validación se calcula el número de errores cometidos en la

clasificación. El error medio para cada par (C, γ) se calcula como la media de los errores obtenidos en cada proceso de entrenamiento-validación. Finalmente se toman como parámetros (C, γ) óptimos aquellos producen el error medio mínimo.

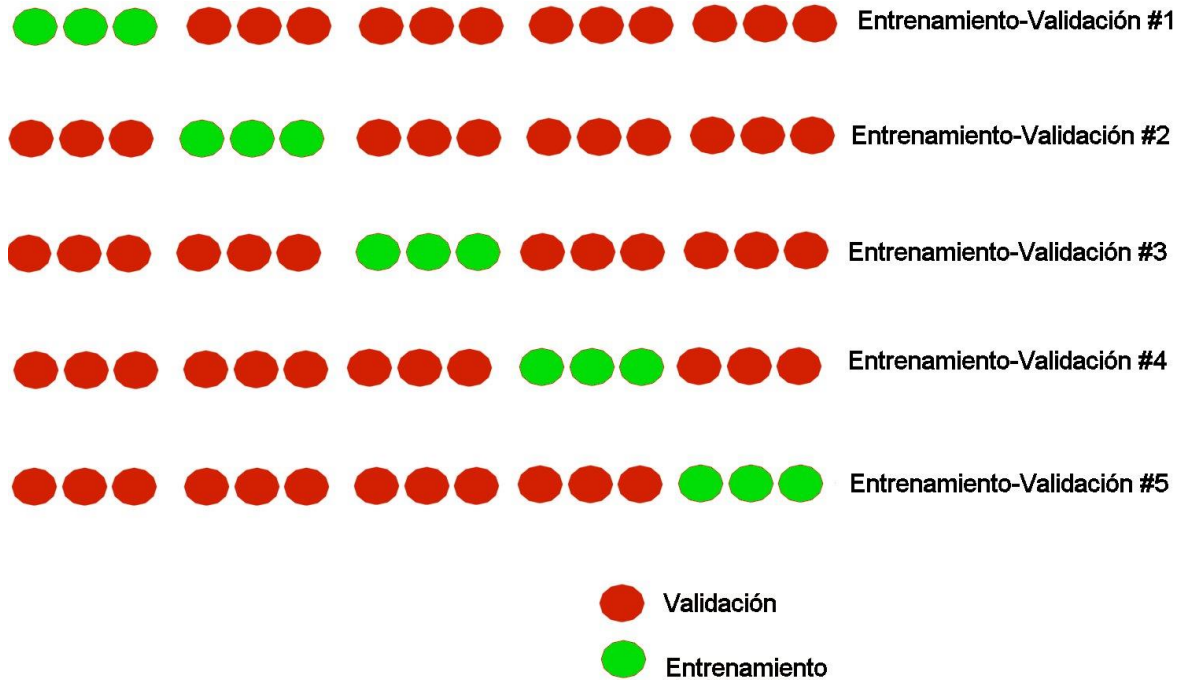


Figura 12: validación cruzada en el entrenamiento de SVM.

Una vez se han obtenido los parámetros óptimos, el SVM se entrena utilizando estos parámetros y el conjunto de imágenes de entrenamiento. A la salida de la etapa de entrenamiento se tiene un modelo que luego es usado para la clasificación de datos nuevos.

Para la clasificación de personas ha sido necesario obtener los parámetros y realizar el entrenamiento de seis SVM. Se han utilizado 480 imágenes de entrenamiento previamente clasificadas a mano. Son imágenes frontales y normalizadas, como las que aparecen en la Figura 13. Estas imágenes forman parte de la base de datos FERET [29], perteneciente a DARPA y al NIST, y que cuenta con un total de 2413 imágenes de 856 individuos, la cual es muy utilizada en el reconocimiento facial. Para cada uno de los seis SVMs se han realizado varias iteraciones del proceso de validación cruzada con el objetivo de realizar en cada iteración un ajuste más fino de los parámetros. Por ejemplo, si en una primera iteración los valores de los parámetros eran:

$$C = \{2, 4, 6, 8\}$$

$$\gamma = \{0.1, 0.2, 0.5, 1.0\}$$

y el error mínimo se ha obtenido para el par (4,0.2) en la siguiente iteración los rangos de parámetros podrían ser:

$$C = \{3, 4, 5\}$$

$$\gamma = \{0.15, 0.2, 0.25, 0.3\}$$

Los valores obtenidos de C y γ en el proceso de optimización de parámetros han sido los siguientes:

- Clasificador 1: $C=4$ $\gamma=0.1$.
- Clasificador 2: $C=30$ $\gamma=0.0244$
- Clasificador 3: $C=1$ $\gamma=0.1$
- Clasificador 4: $C=2$ $\gamma=1$
- Clasificador 5: $C=1$ $\gamma=0.4$
- Clasificador 6: $C=4$ $\gamma=0.4$



Figura 13: Ejemplo de imágenes utilizadas en la fase de entrenamiento.

Una vez obtenidos los parámetros óptimos, cada clasificador es entrenado utilizando estos parámetros. La salida del entrenamiento de cada clasificador es un fichero .xml en el cual se almacenan los parámetros del clasificador y el vector de soporte obtenido durante el entrenamiento y que servirá para clasificar datos nuevos de entrada.

Para implementar la fase de entrenamiento se ha utilizado la librería OpenCV. El código es el siguiente:

```
vector<Mat> images;
vector<int> labels;
//Se leen las imagenes de entrenamiento y sus etiquetas de clasificacion
try {
    read_csv(fn_csv, images, labels);
} catch (cv::Exception& e) {
    cerr << "Error opening file \"" << fn_csv << "\". Reason: " << e.msg << endl;
    exit(1);
}

Ptr<FaceRecognizer> model = createLBPHFaceRecognizer(1,8,8,8);

//Calcula los vectores de características LBP de cada imagen
model->train(images, labels);

//Transforma los vectores de características en un tipo de datos valido para la entrada del SVM
vector<Mat> hist = model->getMatVector("histograms");
Mat histogramas(images.size(),16384,CV_32FC1);
for(int i=0; i<histogramas.rows;i++){
    for(int j=0; j<histogramas.cols; j++){
        histogramas.at<float>(i,j)=hist[i].at<float>(j);
    }
}

Mat clases(labels, true);

//Parametros del SVM
CvSVMParams params;
params.svm_type = CvSVM::C_SVC;
params.kernel_type = CvSVM::RBF;
params.gamma=0.4;
params.C=4;

CvSVM SVM;
SVM.train(histogramas, clases, Mat(), Mat(), params);

//Se guarda el modelo obtenido en el entrenamiento
SVM.save("./svm_mujeres_maduras.xml");
```

Figura 14: Código fase de entrenamiento

Los modelos obtenidos para cada clasificador son los siguientes:

- Clasificador 1: *svm_ninos.xml*
- Clasificador 2: *svm.xml*
- Clasificador 3: *svm_hombres_jovenes.xml*
- Clasificador 4: *svm_hombres_maduros.xml*
- Clasificador 5: *svm_mujeres_jovenes.xml*
- Clasificador 6: *svm_mujeres maduras.xml*

Capítulo 5

5. Resultados.

En este capítulo se detallan los resultados obtenidos con las imágenes de la base de datos, que permitieron entrenar al clasificador (ver Capítulo 4) y con imágenes reales. En este segundo caso, las imágenes se han obtenido de una cámara RGB: el sensor Kinect de Microsoft.

Como se ha descrito en capítulos previos, las imágenes de la cara deben ser frontales para que el sistema funcione correctamente. La Kinect no solo proporciona la cara, sino también los ángulos de inclinación de la misma respecto al sensor. Esto permite descartar automáticamente imágenes y procesar sólo las que el sistema entiende como frontales.

Como se documentará en este Capítulo este proceso de descarte no es totalmente fiable para todos los casos, ya que debido a fallos en la detección de la cara se pueden dar como frontales caras que en realidad no lo son.

5.1.- Resultados obtenidos con imágenes de la base de datos.

Como se ha comentado anteriormente en el apartado primero del capítulo 4 (*Selección de parámetros y entrenamiento*), para realizar el entrenamiento de cada uno de los 6 clasificadores utilizados en este trabajo fin de grado y obtener así los modelos necesarios, se utilizaron imágenes de la base de datos FERET, las cuáles fueron clasificadas a mano y normalizadas, obteniéndose de esta manera un base de datos de imágenes faciales frontales.

Esta base de datos se dividió en 480 imágenes de entrenamiento y 105 imágenes de prueba divididas entre los diferentes grupos de clasificación.

Los resultados obtenidos por cada clasificador con los parámetros óptimos son los que aparecen en la tabla 1.

Clasificador	C	γ	Tasa de aciertos
Svm_niños	4	0.1	89%
Svmsexo	30	0.0244	90%
Svm_hombres_jóvenes	1	0.1	95%
Svm_mujeres_jóvenes	1	0.4	86%
Svm_hombre_maduros	2	1.0	96%
Svm_mujeres_maduras	4	0.4	89%

Tabla 1: Resultados obtenidos con imágenes de la base de datos.

Como podemos apreciar, las tasas de acierto obtenidas son bastante elevadas debido a que se están utilizando imágenes de la misma base de datos con la que se ha realizado el entrenamiento y así, aunque las imágenes de prueba sean diferentes a las de entrenamiento, son parecidas en cuanto a resolución y condiciones de captura.

Una conclusión extraída tras la realización de estas pruebas es que en un 80% los fallos se producen cuando los sujetos tienen una edad próxima a las fronteras de los rangos de clasificación. Es decir, un sujeto de por ejemplo 45 años podría ser clasificado en el rango entre 15 y 40 años, o un sujeto de 17 años, como menor de 15. Es decir, la mayoría de los errores se producen en las fronteras de los rangos de clasificación.

En la figura 15 podemos ver un subconjunto de las imágenes de prueba utilizadas en este apartado.



Figura 15: Imágenes de prueba de la base de datos

5.2.- Resultados obtenidos con imágenes reales a través del sistema.

5.2.1.- Entorno de trabajo y arquitectura del sistema.

Para el desarrollo de las pruebas y obtención de los resultados se ha utilizado un entorno de trabajo formado por los componentes que se describen a continuación:

5.2.1.1. Kinect para Windows.

Kinect para Windows, junto con su SDK (Software Development Kit), permite a los desarrolladores de software crear nuevos programas que interactúan de forma más natural con el PC, haciendo uso de la voz o de gestos manuales y faciales. Haciendo uso de esta SDK es posible la adquisición de las imágenes RGB así como de la imagen de profundidad proporcionada por Kinect. También es posible la detección de hasta seis personas situadas en el campo de visión del sensor y realizar el seguimiento en detalle de dos de ellas. Este seguimiento se realiza colocando un esqueleto sobre los sujetos y calculando la posición de las articulaciones de este modelo de esqueleto. Además permite realizar el seguimiento de la cara de estas dos personas a las que se les ha detectado el esqueleto, dando como salida la posición en 3D de una serie de puntos y su proyección en dos dimensiones, y también la pose de la cara con respecto a los ejes yaw, pitch y roll (ver figura 16).

Este proyecto hace uso de las funcionalidades de detección y seguimiento facial que proporciona la SDK de Kinect para Windows, que nos permite detectar la cara del sujeto en tiempo real, determinar su pose (ángulos de yaw o guiñada, pitch o cabeceo y roll o balanceo (ver figura 16)) y la posición de sus ojos. En caso de que la Kinect detecte dos caras en la imagen, nos quedamos con la del sujeto más cercano.

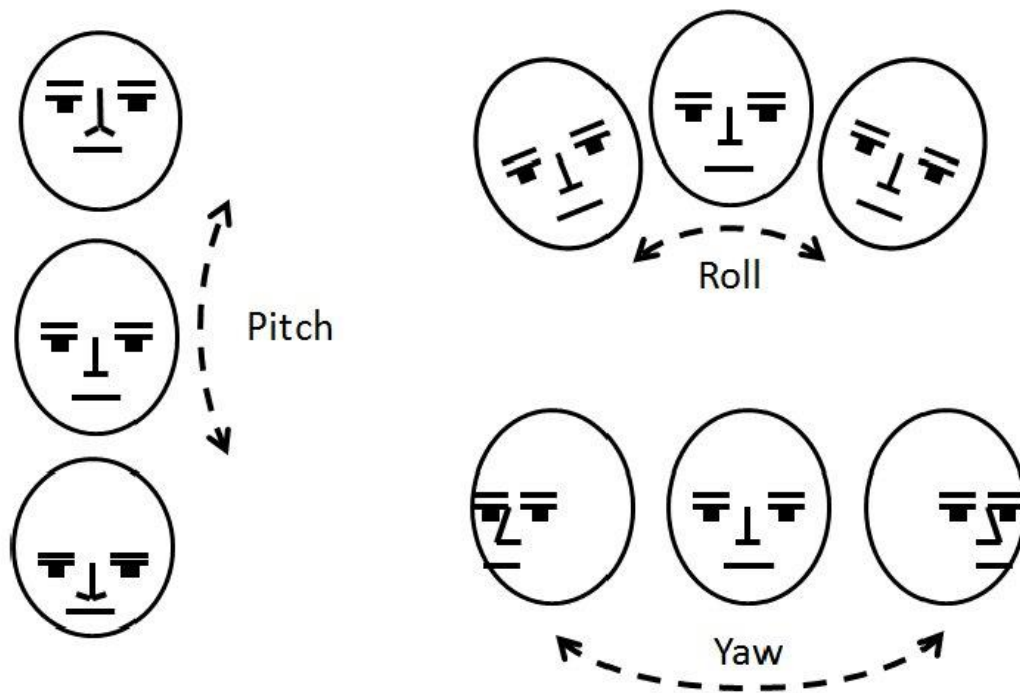


Figura16: Pose de la cara respecto a los ejes.

En este proyecto se ha considerado que una persona está frontal a la Kinect cuando sus ángulos de yaw, pitch y roll se encuentran los tres en el rango (-10, 10) grados.



Figura 17: Imagen de la Kinect V1 para Windows

5.2.1.2. Arquitectura del sistema.

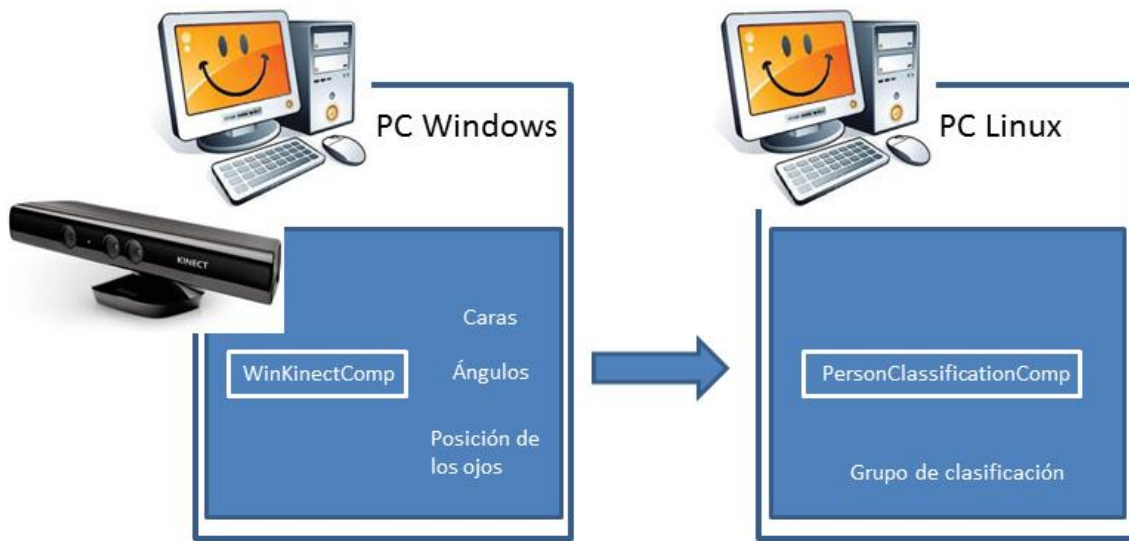


Figura 18: Arquitectura del sistema

El sistema, como aparece en la figura 18, consta de dos módulos o componentes desarrollados utilizando el framework para robótica *Robocomp* explicado en el apartado 5.2.1.4 de este capítulo y que permite la comunicación a través de ICE, de componentes situados en ordenadores diferentes. Uno de los componentes se encarga de la captura de datos desde la kinect y su publicación para poder ser usados por otros componentes. Este componente se denomina WinKinectComp y no ha sido desarrollado en este proyecto, sino que ha sido utilizado como herramienta para la obtención de las imágenes faciales, de los ángulos que determinan la pose de la cara y de la posición de los ojos. En este proyecto se ha desarrollado el otro módulo o componente del sistema, el encargado de la clasificación, denominado PersonClassificationComp. Este componente toma las imágenes faciales y, en primer lugar determina si son frontales, en caso afirmativo utiliza la posición de los ojos para normalizarlas. Tras la normalización, son clasificadas utilizando el algoritmo explicado a lo largo de este trabajo.

5.2.1.3. Ordenadores de sobremesa.

Para la realización de este proyecto se han utilizado dos PC's, uno con sistema operativo Windows para el módulo encargado de la captura de datos desde la Kinect y otro con sistema operativo Linux para el módulo encargado de realizar la clasificación. El PC con

Linux es un AMD Athlon 64 X2 Dual Core 4200+. El PC con Windows es un INTEL Core i5.

5.2.1.4. Robocomp

Robocomp es un framework para generar componentes software para robots que está siendo desarrollado por el laboratorio de robótica de la Universidad de Extremadura en la Escuela Politécnica de Cáceres.

Robocomp es capaz de funcionar en varios sistemas operativos y en un amplio hardware, sin embargo la mayoría de los desarrolladores y testers utilizan Debian o Ubuntu para trabajar, de hecho tienen un **script** para una instalación más sencilla de Robocomp en estos sistemas.

Casi todo el software desarrollado en Robocomp está escrito en **C++** o **Python**, aunque usa el **framework de comunicación** de componentes **ICE** para comunicar entre sí los distintos componentes software que forman Robocomp. ICE es una plataforma middleware ligera y abierta para el desarrollo de aplicaciones basada en objetos distribuidos y que permite reutilizar componentes independientemente del lenguaje en el que esté escrito, además de ser compatible con cualquier plataforma (Windows, Linux, etc.).

Los componentes que proporciona Robocomp están enfocados en software para el control y funcionalidad de los robots, como pueden ser control de servomotores, cálculo de distancias con cámaras estéreo, posicionamiento con ayuda de láser, reconocimiento de objetos e imágenes, sistemas de sonido y reconocimiento de órdenes. Robocomp también proporciona las herramientas necesarias para crear y manejar esos componentes.

5.2.2.- Resultados.

La velocidad media del sistema es de 12 imágenes por segundo y depende del número de clasificadores por los que pase la imagen, es decir, si se clasifica a un niño, el tiempo es menor que si se clasifica a un hombre maduro. El tiempo medido es el del sistema completo, lo que quiere decir que se incluyen la captura y detección de la cara utilizando la Kinect y también el tiempo necesario para la normalización de la imagen facial.

Para evaluar el sistema en condiciones reales, con el entorno de trabajo que se ha explicado anteriormente se han utilizado 10 sujetos de prueba. A cada sujeto se le ha pedido que se posicionara delante de la Kinect, tratando de situar la cara frontal a ella, y se ha procedido a su clasificación. Para realizar esta clasificación se ha aprovechado la ventaja de disponer de toda una secuencia de imágenes del sujeto y no de una sola imagen. Para ello se proporciona como resultado el más repetido en estas 10 imágenes. En caso de haber varios candidatos el sistema proporciona aleatoriamente uno de ellos si éste se repitió cuatro o más veces. En caso contrario, el sistema devuelve el valor indeterminado. Esta regla de decisión fue determinada a priori, no habiéndose modificado en función de los resultados obtenidos.

El proceso de clasificación de cada individuo se ha repetido en veinte ocasiones. En la Tabla 2 se muestran los resultados obtenidos en estas pruebas. La Figura 19 muestra 5 fotogramas consecutivos de cada uno de los sujetos de prueba.

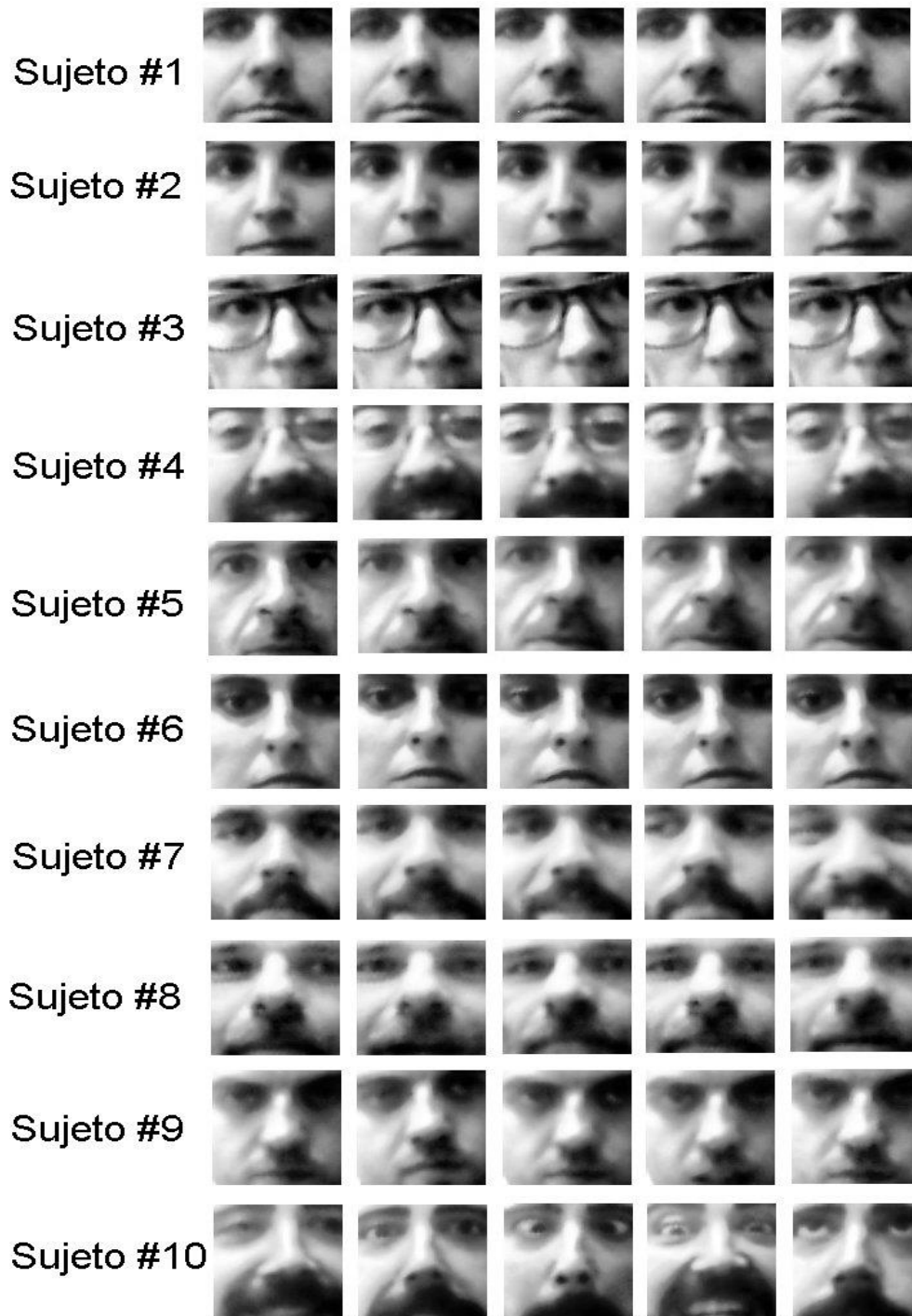


Figura 19: Imágenes de los individuos de prueba

Sujeto	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Correctos	18	16	10	18	17	18	16	15	17	18
Incorrectos	2	4	7	1	3	2	3	3	3	1
Indeterminado	0	0	3	1	0	0	1	2	0	1

Tabla 2. Resultados obtenidos por el sistema en condiciones reales (ver texto)

En total, el porcentaje de acierto para el sistema real es del 81%. De los errores, solo en un 78% de los casos el sistema dio por válido un valor incorrecto. En un 22% de los casos el sistema proporcionó el valor indeterminado. En la Tabla 2 se aprecia como nueve de los diez sujetos han sido clasificados correctamente en más de 15/20 de los casos. Solo el sujeto #3 ofrece unos resultados de clasificación significativamente malos. En la Figura 19 se muestran algunos de los fotogramas asociados a este individuo en concreto. Se aprecia como la cara no se muestra frontal en prácticamente ningún fotograma pese a que la Kinect si las da por frontales. También se aprecia como el área asociada a la cara no está bien definida, y que rasgos significativos de la misma (boca, ojo izquierdo) no siempre aparecen en la misma. Ambos problemas se deben a una mala detección de los puntos significativos de la cara extraídos por la Kinect para realizar el seguimiento facial. Entre éstos puntos se encuentran los ojos, que son utilizados en el proceso de normalización, el cual se realiza de forma incorrecta si los ojos no están bien detectados.

Capítulo 6

En este último capítulo se exponen las conclusiones de todo el trabajo llevado a cabo en este Trabajo Fin de Grado. La mayor parte de estas conclusiones se extraen a partir de los resultados presentados en el capítulo anterior.

En un segundo apartado se exponen las posibles líneas de trabajo futuras con respecto al sistema clasificador que se ha tratado en este trabajo.

6. Conclusiones.

6.1 Conclusiones

A partir de las pruebas realizadas al sistema, podemos extraer como conclusiones que el sistema funciona mejor cuándo se usan imágenes similares a aquellas con las que se ha realizado el entrenamiento, es decir, su funcionamiento es mejor con imágenes de la base de datos que con aquellas capturadas mediante la Kinect. Pero, podemos decir que, tras estudiar su funcionamiento on-line, el sistema parece un buen punto de partida para la realización de un sistema en tiempo real de clasificación de género y edad, el cual puede ser mejorado con algunas propuestas que serán comentadas en el siguiente apartado de este capítulo.

La principal limitación del sistema viene impuesta por el hecho de necesitar que las imágenes de entrada al sistema sean frontales a la cámara. Esta limitación viene derivada del uso de únicamente patrones frontales de entrenamiento. Esto simplifica el proceso de entrenamiento, siendo necesarias un número mucho menor de imágenes que si se utilizaran patrones para cada una de las posibles poses de la cara (para ser coherentes con lo realizado anteriormente se necesitan 120 imágenes por pose). Además, evita también el hecho de tener que determinar qué poses considerar y cuáles no. Sin embargo, esta simplificación hace que el sistema no sea invariante a variaciones en la pose de la cara, es decir, que únicamente funcione correctamente cuando el sujeto a clasificar se encuentra frontal a la cámara.

Otra limitación del sistema es la necesidad de un proceso previo de normalización de las imágenes de entrada, esto se debe al uso de LBP como caracterizador de las imágenes. Para que este proceso se realice correctamente es necesario estimar de forma correcta la posición de los ojos en las imágenes. Un error en esta estimación da lugar a normalizaciones incorrectas que provocan que el sistema clasifique de forma incorrecta.

6.2 Trabajo futuro

Para conseguir que el sistema sea invariante a la pose de la cara, se podría entrenar el sistema con patrones de cada pose. Esto, como se ha comentado anteriormente, presenta otro tipo de problemas, como la gran cantidad de imágenes de entrenamiento necesarias. Otra solución sería realizar un preprocesado previo de las imágenes capturadas por la Kinect, realizando una rectificación de las imágenes de forma que queden frontales. Esto se puede realizar haciendo uso de la SDK de Kinect, la cual proporciona funciones para seguimiento de caras en 3D. Estas funciones calculan la posición 3D de una serie de puntos de la cara, así como su proyección en la imagen 2D. Además calculan la pose global de la cara con respecto a un sistema de coordenadas de referencia. Los puntos 3D estimados se pueden usar para calcular una malla triangular a la cual se le puede asociar una textura extraída de la imagen facial, teniendo como resultado un modelo 3D de la cara. Este modelo se puede transformar hasta situarlo frontal a la cámara. Una vez situado frontal a la cámara se puede generar una proyección 2D del modelo, lo que nos da una imagen frontal de la cara del sujeto.

REFERENCIAS.

- [1] Jain, Anil K.; Ross, Arun (2008). "Introduction to Biometrics". In Jain, AK; Flynn; Ross, A. Handbook of Biometrics. Springer.
- [2] Yael Adnin, Yael Moses and Shimon Ullman, "Face recognition: The problem of compensating for changes in illumination direction", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, 1997, pp712-732.
- [3] Gupta, S.; Markey, M. K.; Bovik, A. C. (2010). "Anthropometric 3D Face Recognition". International Journal of Computer Vision (IJCV) 90 (3): 331–349.
- [4] A. Rashad, A Hamdy, M A Saleh and M Eladawy, "3D face recognition using 2DPCA," (IJCSNS) International Journal of Computer Science and Network Security, Vol.(12),2009
- [5] A. J. Goldstein, L. D. Harmon, and A. B. Lesk, "Identification of Human Faces," Proc. IEEE, May 1971, Vol. 59, No. 5, 748-760.
- [6] L. Sirovich and M. Kirby (1987). "Low-dimensional procedure for the characterization of human faces". Journal of the Optical Society of America A 4 (3): 519–524.
- [7] M. Turk and A. Pentland (1991). "Face recognition using eigenfaces". Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 586–591.
- [8] Greene, Lisa (2001-02-15). "Face scans match few suspects" (SHTML). St. Petersburg Times. Retrieved 2011-06-30.
- [9] W. Zhao, R. Chellappa, A. Rosenfeld and P.J. Phillips. Face Recognition: A literature survey.
- [10] <http://www.technologyreview.com/news/513901/how-facial-recognition-tech-could-help-trace-terrorism-suspects/>

- [11] Locating facial features for age classification. Kwon, Y. and Lobo, N. V. 1993. Proceedings of SPIE - the Intl. Society for Optical Engineering. Vol. 2055, pp. 62-72.
- [12] Age Classification from Facial Images. Kwon, Y. and Lobo, N. V. 1, Abril 1999, Computer Vision and Image Understanding, Vol. 74, pp. 1-21.
- [13] Alley, T. R. Social and Applied Aspects of Perceiving Faces. s.l. : Erlbaum Assoc., 1988.
- [14] Active Appearance Models. Cootes, T. F., Edwards, G. J. and Taylor, C. J. 1998, ECCV.
- [15] Automatic Age Estimation Based on Facial Aging Patterns. Geng, X., Zhou, Z. H. and Smith- Miles, K. 12, Diciembre 2007, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, pp. 2234-2240.
- [16] A. Gunay, and V. Nabyev, "Automatic age classification with LBP", Proc. Int'l. Symp. Computer and Informatic Sciences, 2008.
- [17] A. Lanitis, C. Draganova, and C. Christodoulou, "Comparing different classifiers for automatic age estimation", IEEE Trans. Systems, Man and Cybernetics Part B, vol. 34, no. 1, pp. 621-628, Feb. 2004.
- [18] G. Guo, Y. Fu, T.S. Huang, and C. Dyer, "Locally adjusted robust regression for human age estimation", Proc. IEEE Workshop Applications of Computer Vision, 2008.
- [19] G. Guo, Y. Fu, C. Dyer, and T.S. Huang, "Image-based human age estimation by manifold learning and locally adjusted robust regression", IEEE Trans. Image Processing, vol. 17, no. 7, pp. 1178-1188, July 2008.

- [20] A. Lanitis, C. Taylor, and T. Cootes, "Toward automatic simulation of aging effects on face images", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 442-455, April 2002.
- [21] Y. Fu, Y. Xu, and T.S. Huang, "Estimating human ages by manifold analysis of face pictures and regression on aging features", *Proc. IEEE Conf. Multimedia and Expo*, pp. 1383-1386, 2007.
- [22] Y. Fu And T.S. Huang, "Human age estimation with regression on discriminative aging manifold", *IEEE Trans. Multimedia*, vol. 10, no. 4, pp. 578-584, June 2008.
- [23] G. Guo, Y. Fu, T.S. Huang, and C. Dyer, "Locally adjusted robust regression for human age estimation", *Proc. IEEE Workshop Applications of Computer Vision*, 2008.
- [24] Guo, Y. Fu, C. Dyer, and T.S. Huang, "Image-based human age estimation by manifold learning and locally adjusted robust regression", *IEEE Trans. Image Processing*, vol. 17, no. 7, pp. 1178-1188, July 2008.
- [25] G. Guo, Y. Fu, T.S. Huang, and C. Dyer, "Locally adjusted robust regression for human age estimation", *Proc. IEEE Workshop Applications of Computer Vision*, 2008.
- [26] G. Guo, Y. Fu, C. Dyer, and T.S. Huang, "Image-based human age estimation by manifold learning and locally adjusted robust regression", *IEEE Trans. Image Processing*, vol. 17, no. 7, pp. 1178-1188, July 2008.
- [27] G. Guo, Y. Fu, T.S. Huang, and C. Dyer, "A probabilistic fusion approach to human age prediction", *Proc. IEEE Computer Vision and Pattern Recognition-Semantic Learning and Applications in Multimedia Workshop*, 2008
- [28] H.-C. Lian, and B.-L. Lu, "Multi-view gender classification using local binary patterns and support vector machines", *Proc. 3rd Int'l. Symp. on Neural Networks*, vol. 2, pp. 202-209, 2006.

[29] P. Jonathon Phillips, Harry Wechsler, Jeffrey Huang, Patrick J. Rauss: The FERET database and evaluation procedure for face-recognition algorithms. *Image Vision Comput.* 16(5): 295-306 (1998)

[30] A comparison of techniques for robust gender recognition. Rojas-Bello, R.N., Lago-Fernandez, L.F., Martínez-Munoz, G. y Sánchez-Montañés, M.A. 2011. *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP)*. pp. 561-564.